

Express Mail Label No. EV 2270 876 90 US

**SYSTEM AND METHOD FOR SYNCHRONIZING DATA IN A NETWORKED  
SYSTEM**

Inventors:  
**John Ezell**  
**Neil Gilmartin**  
**Stephen Fitzpatrick**

John Buckert  
Cantor Colburn LLP  
201 W. Big Beaver Rd., Suite 370  
Troy, Michigan 48084  
Telephone: 248-524-2300

Attorney Reference No.: BLL-0119  
Bell South Number: 030366

## SYSTEM AND METHOD FOR SYNCHRONIZING DATA IN A NETWORKED SYSTEM

### FIELD OF INVENTION

[0001] The invention relates to a system and method for synchronizing data in a first computer server and a second computer server in a networked system.

### BACKGROUND OF INVENTION

[0002] Networked systems generally utilize numerous computer servers that communicate with one another. Further, known networked systems have utilized redundant copies of data sets or databases to allow easy access thereto at multiple physical locations. Generally, a primary computer server will have a primary or “master” plurality of data sets which contains the most “up to date” data sets. For purposes of discussion, a data set is a logically grouped set of software attributes associated with an entity that can be electronically stored in a computer memory. An entity may comprise, for example, one of the following: a communication network element, data, a software event, or a logical operator. A secondary computer server often will have a redundant copy of the primary data set or a copy of a portion of the primary data set, which is called a secondary data set.

[0003] When supporting the known networked system it is necessary to synchronize the primary plurality of data sets and the secondary plurality of data sets to ensure that particular attributes stored in the predetermined data sets match one another. To accomplish this system synchronization task, known methods transmit the entire plurality of primary data sets from the first computer server to the secondary computer server to compare and update the secondary plurality of data sets, or vice versa. Accordingly, during this system-synchronization task, the transmission of a relatively large number of data sets, each having a relatively large number of attributes, between the servers in a networked system adversely affects concurrent transmission of other data

because most of the system communication bandwidth is being utilized to synchronize the data sets.

[0004] Accordingly, there is a need for a system and method for performing data synchronization in a networked system that reduces the amount of information transmitted between the primary and secondary computer servers.

## SUMMARY OF INVENTION

[0005] The foregoing problems and disadvantages are overcome by a system and method for synchronizing data in a first computer server and a second computer server in accordance with the exemplary embodiments disclosed herein.

[0006] A method for synchronizing data in first and second computer servers in accordance with exemplary embodiments is provided. The first computer server includes a first plurality of data sets each having a first identifier and a first set of attributes. The second computer server includes a second plurality of data sets each having a second identifier and a second set of attributes. The method includes transmitting a second identifier and a second checksum value both associated with one of the second plurality of data sets to the first computer server. The method further includes accessing one of the first plurality of data sets having a first identifier corresponding to the transmitted second identifier to determine a first checksum value associated with the accessed data set. Finally, the method includes transmitting the one of the first plurality of data sets from the first computer server to the second computer server to replace a second set of attributes of the one of the second plurality of data sets with the first set of attributes of the one of the first plurality of data sets when the first checksum value is not equal to the transmitted second checksum value.

[0007] A networked system in accordance with exemplary embodiments utilizes a first computer server and a second computer server operably communicating with one another. The first computer server is configured to store a first plurality of data sets each having a first identifier and a first set of attributes. The second computer server is

configured to store a second plurality of data sets each having a second identifier and a second set of attributes. The second computer server is further configured to transmit a second identifier and a second checksum value both associated with one of the second plurality of data sets to the first computer server. The first computer server is further configured to access one of the first plurality of data sets having a first identifier that corresponds to the transmitted second identifier to determine a first checksum value associated with the accessed data set. The first computer server is further configured to transmit the one of the first plurality of data sets from the first computer server to the second computer server to replace a second set of attributes of the one of the second plurality of data sets with the first set of attributes of the one of the first plurality of data sets when the first checksum value is not equal to the transmitted second checksum value.

[0008] Other systems, methods and/or computer program products according to the embodiments will be or become apparent to one with skill in the art upon review of the following drawings and detailed description. It is intended that at all such additional systems, methods, and/or computer program products be within the scope of the present invention, and be protected by the accompanying claims.

## BRIEF DESCRIPTION DRAWINGS

[0009] Figure 1 is a block diagram of a networked system having a primary computer server and a secondary computer server.

[0010] Figure 2 illustrates an exemplary data set in a primary computer server and an exemplary data set in a secondary computer server.

[0011] Figures 3A-3C illustrate a flowchart of a method for synchronizing data in first and second computer servers.

## DESCRIPTION OF AN EMBODIMENT

[0012] Referring to the drawings, identical reference numerals represent identical components in the various views. Referring to Figure 1, networked system 10 includes a primary computer server 12 and a secondary computer server 22. As will be discussed in greater detail below, system 10 and a method related thereto can synchronize data sets in computer servers 12, 22 while decreasing the amount of information exchanged between the servers.

[0013] Networked system 10 can be any type of known network including, but not limited to, a public switched telephone network (PSTN), a wide area network (WAN), a local area network (LAN), a global network (e.g. Internet), a voice over Internet protocol (VOIP) network, a virtual private network (VPN), and an intranet. Networked system 10, maybe implemented using a wireless network or any kind of physical communication network known to those skilled in the art.

[0014] Primary computer server 12 is provided to communicate with secondary computer server 22 and network elements 32, 34, 36. Computer server 12 is further provided to store a “master” data set that contains the most up-to-date data utilized in system 10. Server 12 includes a microprocessor 14 communicating with various computer readable storage medium. The computer readable storage media preferably includes nonvolatile and volatile storage in a read-only memory (ROM) 16 and a random access memory (RAM) 18. The computer readable medium may be implemented using any of a number of known memory devices such as PROMs, EPROMs, EEPROMS, flash memory or any other electric, magnetic, optical or combination memory device capable of storing data, some of which represent executable instructions used by microprocessor 14. Microprocessor 14 may communicate with various servers and network elements via an input/output (I/O) interface 20.

[0015] Secondary computer server 22 is provided to communicate with primary computer server 12 and to store a copy of the entire “master” plurality of data sets or a

portion of the “master” plurality of data sets stored in server 12. Server 22 may also be utilized to perform diagnostic analysis of the network 10 using the data set stored in server 22. Server 22 includes a microprocessor 24, ROM 26, RAM 28, and I/O interface 30. As illustrated in Figure 1, I/O interface 30 of server 22 may be connected to I/O interface 20 of server 12 for communication therebetween. Further, network elements 32, 34, 36 maybe connected to I/O interface 20 and server 12 for communication therebetween. It should be noted that each of computer servers 12, 22 could be implemented using a multi-processor server architecture.

[0016] Network elements 32, 34, 36 may comprise any device that can communicate over network 10. For example, network elements 32, 34, 36 may comprise multiplexer units (having physical racks, shelves, communication cards, microprocessors, and communication ports) that communicate with other servers or other network elements (not shown).

[0017] Referring to Figure 2, the plurality of data sets used in primary computer server 12 and secondary computer server 22 will be explained. Blocks 42, 44 correspond to a plurality of data sets stored in the memory of computer server 12, hereinafter referred to as the "first plurality of data sets". Block 46 corresponds a plurality of data sets stored in the memory of computer server 22, hereinafter referred to as the "second plurality of data sets". Block 48 corresponds to intermediate calculations performed in computer server 12 to calculate a checksum associated with a particular identifier which will be explained in greater detail below.

[0018] Before engaging in a detailed discussion for synchronizing data between the first and second plurality of data sets in computer servers 12, 22, respectively, a general overview will be provided. In order to synchronize data between the first and second plurality of data sets in computer servers 12, 22, the servers will utilize: (i) a predetermined entity identifier for each related data set in computer servers 12, 22 that can be recognized by both servers, (ii) each data set will have a set of attributes (e.g., fields of data) of common interest and a corresponding checksum associated with the

attributes, and (iii) the entity identifier and each attribute of each data set will have a predetermined format type when transmitting the data set from computer server 12 to computer server 22.

[0019] Referring again to blocks 42, 44, the first plurality of data sets may be partitioned into two or more memory locations in memory. Further, each of the first plurality of data sets has an entity identifier (e.g., E1) and attributes (e.g., a, d, checksum1, f, b, c) associated with the identifier. Each entity identifier identifies one data set in a computer server and further corresponds to a predetermined entity. As discussed above, an entity may comprise, for example, one of the following: a communication network element, data, a software event, or a logical operator. Further, each entity identifier has an associated checksum value based on a checksum calculation performed on one or more of attributes associated with the entity identifier. For example, entity identifier (E1) may include an attribute (checksum1) corresponding to a checksum value calculated using attributes (a, c) of entity (E1). Further, attributes associated with an entity identifier may have a data format types different from one another. For example, entity identifier (E1) has attributes (a, d, checksum1) having format types of (B1, B2, Y3), respectively.

[0020] Referring to block 46, each data set of the second plurality of data sets also includes an entity identifier (e.g., E1) and attributes (e.g., a, c, checksum1) associated with the identifier. Further, each entity identifier has a checksum value based on a checksum calculations performed on one or more of attributes associated with the entity identifier. For example, entity identifier (E1) includes an attribute (checksum1) corresponding to a checksum value calculated using attributes (a, c) of entity (E1) in block 46. Further, attributes associated with an entity identifier may have a predetermined data format types different from one other. For example, entity identifier (E1) of block 46 has attributes (a, c, checksum1) having format types of (A1, A2, Y3), respectively.

[0021] Referring to block 48, this block represents a temporary data buffer in RAM 18 of computer server 12 for formatting attributes of data sets to predetermined format types for: (i) transmitting data sets between computer servers 12, 22, and (ii) calculating checksum values associated with predetermined data sets that are subsequently stored in the memory of server 12. For example, data buffer 48 can be used for converting attributes (a, c) of entity (E1) in block 42 from format types (B1, B3) to format types (Y1, Y2) to calculate the (checksum1) value based on attributes (a, c). The contents of data buffer 48 can be transmitted to computer server 22 or copied to data blocks 42, 44 of computer server 12. Although one data buffer 48 is shown for a computer server 12, it should be noted that computer server 22 also utilizes a similar temporary data buffer for converting attributes of data sets to predetermined formats and for calculating checksum values.

[0022] Referring to Figures 3A-3C, a method for synchronizing the first plurality of data sets and the second plurality of data sets in computer servers 12, 22, respectively, will now be explained. For purposes of simplicity, the method will be explained assuming one data set identified by entity identifier (E1) in blocks 42, 44 of computer server 12 will be used to synchronize a related data set identified by identifier (E1) in block 46 of computer server 22. Further, it will be assumed that only the common attributes (a, c) of entity identifiers (E1) will be synchronized.

[0023] At step 60, primary computer server 12 updates one or more checksum values associated with the first plurality of data sets of blocks 42, 44. To implement step 60, steps 64-70 will be performed. At step 64, computer server 12 modifies an attribute of one data set of the first plurality of data sets stored in server 12. For example, referring to Figure 2, computer server 12 may modify attributes (a, c) of entity identifier (E1) of blocks 42, 44. Thus, at this time the attributes (a, c) of entity identifier (E1) in blocks 42, 44 will not be synchronized or match attributes (a, c) of entity identifier (E1) in block 46 of computer server 22.

[0024] Next at step 66, computer server 12 sets a checksum value associated with the modified data set equal to zero. For example, computer server 12 can set the (checksum1) value of entity identifier (E1) of block 42 equal to zero.

[0025] Next at step 68, computer server 12 accesses the data set having a checksum value equal to zero and formats one or more of the attributes associated with the data set to predetermined formats types. For example, computer server 12 can access entity identifier (E1) of block 42 having a (checksum1) value equal to zero. Thereafter, referring to block 48, computer server 12 can convert attributes (a, c) associated with entity identifier (E1) to data formats (Y1, Y2).

[0026] Next at step 70, computer server 12 calculates a checksum value associated with the accessed data set based on the formatted attributes. For example, computer server 12 can calculate a new (checksum1) value in block 48 based on the attributes (a, c) having data formats (Y1, Y2). Thereafter, the updated attributes (a, c) and new (checksum1) of the identifier (E1) can be stored in the memory of computer server 12.

[0027] Referring to Figures 2, 3A, 3C, after step 60 the method advances to step 62. At step 62, one or more of the second plurality of data sets in the secondary computer server 22 is synchronized with one or more of the first plurality of data sets in the primary computer server 12. To implement step 62, steps 72-80 are performed.

[0028] At step 72, secondary computer server 22 transmits a second entity identifier and a second checksum value both associated with one of the second plurality of data sets to the computer server 12. For example, computer server 22 can transfer identifier (E1) and (checksum1) of block 46 to computer server 12.

[0029] Next at step 74, computer server 12 accesses one of a plurality of data sets having a first identifier corresponding to the transmitted second identifier to obtain a first checksum value associated with the accessed data set. For example, computer server 12 can utilize a received entity identifier (E1) from computer server 22 to access the data set

identified by entity identifier (E1) of block 42 to obtain the associated (checksum1) value.

[0030] Next at step 76, computer server 12 makes a determination as to whether the (checksum1) value of block 42 is not equal to the received (checksum1) value of block 46. If the value of step 76 is true, indicating that the two data sets are not synchronized, the method advances to step 78. Otherwise, the method advances to step 82.

[0031] At step 78, computer server 12 transmits the accessed one of the first plurality of data sets to secondary computer server 22. For example, computer server 12 can convert attributes (a, c) of entity identifier (E1) of block 42 from format types (B1, B3) to format types (Y1, Y2) in block 48 and then transfer the entity identifier (E1) and formatted attributes (a, c) to computer server 22.

[0032] Next at step 80, computer server 22 replaces a second set of attributes of the one of the second plurality of data sets with the first set of attributes of the transmitted one of the first plurality of data sets. For example, computer server 22 can replace the attributes (a, c) of entity identifier (E1) of block 46 with updated attributes (a, c) of entity identifier (E1) from blocks 42, 44.

[0033] Next, at step 82, computer server 12 determines whether there is more data to synchronize in computer servers 12, 22. If the value of step 82 equals "yes", the method advances to step 72. Otherwise, the method is ended.

[0034] Although the inventive method was explained by synchronizing one data set in each of computer servers 12, 22, it should be understood that the method could be iteratively performed to synchronize a plurality of data sets in computer server 12 with a plurality of data sets in computer server 24.

[0035] As discussed above, embodiments for synchronizing data information in first and second computer servers in a networked system provide substantial advantages

over known systems. In particular, the embodiments only transmit an entity identifier and a checksum value and associated with a data set from a first server to a server to determine whether the data sets are synchronized, instead of transmitting the entire data set from the first computer server to the second computer server. Thus, the embodiments substantially reduce the amount of information that is transmitted through the networked system during synchronization of the data sets in each of the computer servers.

[0036] While the invention is described with reference to an exemplary embodiment, it will be understood by those skilled in the art that various changes may be made and an equivalence may be substituted for elements thereof without departing from the scope of the invention. In addition, many modifications may be made to the teachings of the invention to adapt to a particular situation without departing from the scope thereof. Therefore, it is intended that the invention not be limited to the embodiment disclosed for carrying out this invention, but that the invention includes all embodiments falling within the scope of the intended claims. Moreover, the use of the terms first, second, etc. does not denote any order of importance, but rather the terms first, second, etc. are used to distinguish one element from another.